WHAT'S YOUR FOOD?

*

Siddharth Itagi
CSE DEPT
PES UNIVERSITY
Bengaluru, India
siddharth.siya@gmail.com

Vaibhav Pundir
CSE DEPT
PES UNIVERSITY
Bengaluru, India
vaibhav.pundir97@gmail.com

Krishna Khurana
CSE DEPT
PES UNIVERSITY
Bengaluru, India
krishnadkhurana@gmail.com

***Abstract*—Our project aims to classify the recipes based on its ingredients into its respective cuisine. We began this journey firstly by pre-processing the data. This was indeed required as the data wasn't clean.**
After this we performed summary statistics in order to generate and develop insights about our data. This was followed by building several feature sets and then trying various classification algorithms in order to obtain the highest accuracy of 79.2%.

I. INTRODUCTION

In today's generation finding a recipe with the required ingredients is every person's cup of tea.Well everything is available on the greatest discovery of humankind ”THE INTERNET”. But doing the same in the other way does not seem to be that simple. Classifying a cuisine based on its ingredients? This is really some food for thought.

Always wondered which cuisine does the delicious food that you eat belong to? Does just knowing the ingredients suffice to your answers? Well if not earlier(even if you were able to earlier but were not satisfied with the results) now it will, as we will make this possible. We are basically aiming at predicting a recipes cuisine based on its ingredients. We will just want to know the ingredients and there it is, the name of the cuisine is in front of you in no time.

Well you may be thinking how is this of any help in practicality? Suppose you are a cook of a restaurant and you are asked to prepare an Indian cuisine from the ingredients available to you, you would firstly fall into a dilemma as to which ingredients do actually make up the Indian cuisine. But with the help of our technique this wont be a problem anymore.

Our project will also help determine which are the most frequently occurring ingredients for a particular cuisine. This will help users get an insight about the presence of an ingredient and directly determine the cuisine from it. For example, the presence of Garam Masala in a particular recipe should give an intuition to a user that this particuar recipe will tend to be more inclined towards being an Indian cuisine. Our team aims to achieve the features stated above and hence contribute towards making the not so easy task until now a much easier one.

II. PREVIOUS WORK

There has been some reasonable work done on this before. 4. A very brief summary of the literature survey report is mentioned below.

A. XGBOOST

The paper[5] mentions about using Random forests using SKLearn in Python and XG Boosting algorithm using R. They used Random Forests with a set of 200 trees. Using XG Boost they reported an accuracy of 80.4% while Random Forests gave an accuracy of 76.4%.

B. SVM and Logistic Regression

[2]The dataset was divided into 2 sets, 80% for training and 20% for validation. To reduce the number of features they used Bag Of Words model and removed the words of length less than 3.They also used PCA to further reduce the features but that wasn't of any help to the accuracy.
They have used different classification techniques like Naive Bayes, SVM, Random Forest, Logistic Regression, KNN and Decision Trees. Accuracy of 78% using SVM and Logistic Regression, 73% with Random Forest and Naive Bayes and around 60% using KNN and Decision Trees were reported.

C. Ensemble of SGD Classifier and Random Forests

[1] By careful investigation of data, it was found that some words had accents and some words were present in different forms. This problem was overcome by scikit library of python. The dataset was transformed into a Bag-Of-Words model to feed it into a Machine Learning model. Here finally Voting Classifiers were used, the sub-classifiers were SGD Classifier(weight 3) and Random Forests(weight 1) which gave outstanding accuracy.

D. Multinomial Logistic Regression

In this paper [3] they have mentioned about using four classification techniques for building. The model used were Naive Bayes Classifier, Multinomial Logistic Regression, Random Forest Classifier and Linear Support Vector Classification

(SVC). To pre-process the data WordNetLemmatizer of NLTK library was used. The raw textual data was converted into vectors with the help of TfidfVectorizer. The accuracy of the various models were reported as 73.5% for Naive Bayes, 76.2% for Random Forests, 79% for Linear SVC and 78.8% for Multinomial Regression.

III. PROBLEM STATEMENT

A. *Overview*

Our problem statement defines to predict the cuisine based on ingredients of a recipe. It is a multi-classification problem. Our data being labelled i.e. the value of the predictor variable was already present, we resorted to Supervised Learning methods.

B. *Dataset*

We obtained our dataset from a competition hosted by Kaggle titled ”What’s Cooking” (https://www.kaggle.com/c/whats-cooking/data). Our dataset consists of 39774 rows x 3 columns. The 3 attributes of the data-set are ID, Cuisine and Ingredients.
The data-set spans over 20 cuisines. The attribute ”Ingredients” is basically a list of list containing ingredients.

IV. OUR APPROACH

A. *PREPROCESSING THE DATA*

Since our data was textual, it needed a lot of cleaning and preprocessing. By preprocessing we mean making the data suitable and appropriate in order to proceed further with our model. This is one of the most important step in building a correct and accurate model.

We started by first checking for missing values in the dataset. Our dataset didn’t consist of any missing values. We have then converted all the ingredients in the dataset into lowercase so as to maintain uniformity over the entire dataset. We have then used nltk corpus to remove the occurrences of stop words in the ingredients. Then from the String library, punctuation was used to remove any punctuation present in ingredients.

After this we used lemmatization in order to make the ingredients consistent over the entire dataset. This was indeed an essential requirement due to the presence of a word and its plural form which convey the same meaning in this context and can thus affect our predictions in the future. For example, egg and eggs, banana and bananas, tomato and tomatoes, etc.

Later, we have converted this into a new dataframe which has the cleaned and pre-processed data. (Note that this dataframe is used in further steps)

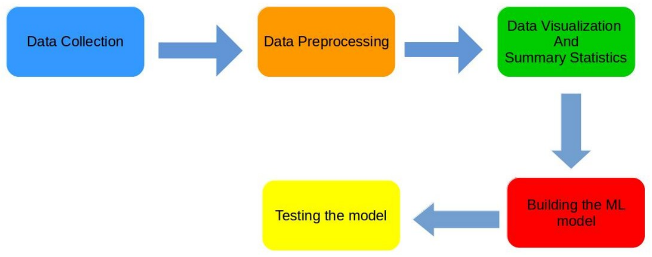

Fig. 1. Block diagram of our approach

B. *DATA VISUALIZATION AND SUMMARY STATISTICS*

Visualization is another important part in building a model as it gives us a clear insight as to how the data is actually distributed. This helps us to make conclusions and proceed in the right direction for further analysis.

We first tried to find the most frequent ingredients in the dataset. For this we used a donut chart which is able to correctly depict the top ingredients in the data. We found ”salt” to be the most frequent ingredient in the entire data. Well this seems to be reasonable actually because ”salt” is actually an essential ingredient in most recipes.

We also tried to find out the occurrences of all the cuisines in this dataset. We used a barplot for this which clearly depicts the same. We found out that ”Italian” is the most common cuisine in this data.

Every cuisine, although having many common ingredients with others, uses a small number of ingredients that combine for giving unique flavors. So which are those ingredients that belong to a specific cuisine? In order to find that, the top 5 ingredients in each cuisine in terms of frequency were visualized. It is clear to see that in most cuisines salt is the main seasoning ingredient while soy sauce and fish sauce takes the top spot in Asian cuisines. However, these most used ingredients are the general ones. Salt is used in almost every recipe of every cuisine. So does Soy Sauce. So how do you find out which ingredients belong to specific cuisines?
To find this out, the relative usage of each ingredients in each cuisine is calculated. To calculate the relative usage of an ingredient, the percentage of how many times that ingredient appears in a cuisine is divided by the its usage among all other cuisines.
With this we are able to see the results that we wanted. The most cuisine specific ingredients are shown with the help of a horizontal bar chart with ingredients on vertical axis and its relative frequency on x-axis.

C. *SPLITTING OF THE DATASET*

In order to validate the results of the various classification algorithms we had to split the data into training and testing data. This was done in the ratio of 30:70 for testing and training respectively.

D. *BUILDING THE MODEL*

Since our data is categorical we need to transform these categorical values to numerical values in order to make it suitable for applying various classification techniques. We have used different approaches for this categorical to numerical conversion.

The approaches that we used were:
1) Tf-idf approach
2) Count vectorizer approach
3) Doc2Vec Approach

Once these transformations have been performed and now that the data was in the suitable form to be trained, we applied various classification techniques. Some of the classification techniques we tried were:
1) SVM (Support Vector Machine)
2) Naive Baye's Classifier
3) Random Forests
4) XG Boost
5) Logistic Regression
6) Neural Network(MLP CLassifier)

Finally, the results that were obtained from these various classification techniques were validated again the testing dataset.

V. EXPERIMENTAL STUDY AND ANALYSIS

As mentioned above we tried to use various categorical to numerical conversion techniques. These techniques followed by model implementation have been explained below in great detail.

1) TF-IDF Approach: Term frequencyInverse document frequency, is a statistic used to reflect how important a word is to a document in a collection or corpus. The intuition behind this is that if a particular word occurs multiple times in a document, its relevance should be more meaningful than other words that appear fewer times (TF). At the same time, if a word occurs many times in a document but is also present in many other documents, it maybe because this word is just a frequent word and not because it is relevant or meaningful (IDF).

This was implemented by first converting and splitting all the words into documents. This was an essential part so as to be able to determining the tf-idf scores. The calculation of the scores is done by the tfidf vectorizer which is a module present in sklearn.feature_extraction.text. Later this was converted into a form suitable for it to be passed as a parameter for various classification techniques.

The first approach that we started with was SVM(Support

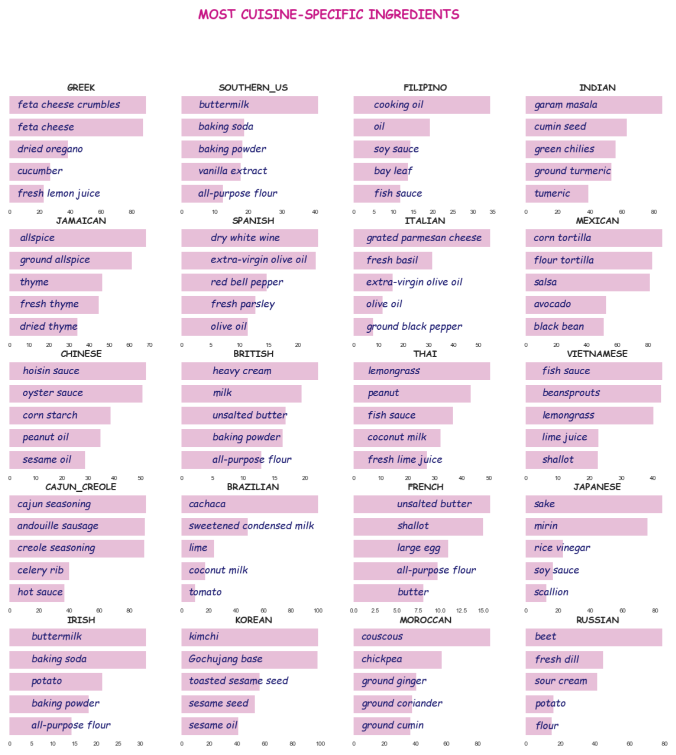

Fig. 2. Most Cuisine Specific Ingredients

Vector Machines). The first question that arises is that why start with SVM? Well, SVM is one of the best classification algorithms. This is due to its ability to capture complex data points without having to perform difficult transformations. We used a linear kernel for SVM.We implemented SVM using sklearn.svm library in python Finally we were able to achieve an accuracy of 78.3%.

We then tried implementing Naive Baye's Classifier, Random Forests and XG Boost algorithm. An accuracy of 65.8% ,75.0% and 73.17% were reported respectively for the above models.

2) Count Vectorizer Approach: In the Bag-OfWords (BoW) model we assign each word a unique number.So now any document we see can be encoded as a fixed-length vector with the length of the vocabulary of known words. The value in each position in the vector could be filled with a count or frequency of each word in the encoded document.

One of the methods provided by scikit-learn library is CountVectorizer. With the help of CountVectorizer we tokenize a collection of text documents and build a vocabulary of known words, but also to encode new documents using that vocabulary.After vectorizing the text, it was transformed into a matrix form which can be fed to various classification algorithms.

Random Forests builds multiple decision trees and merges

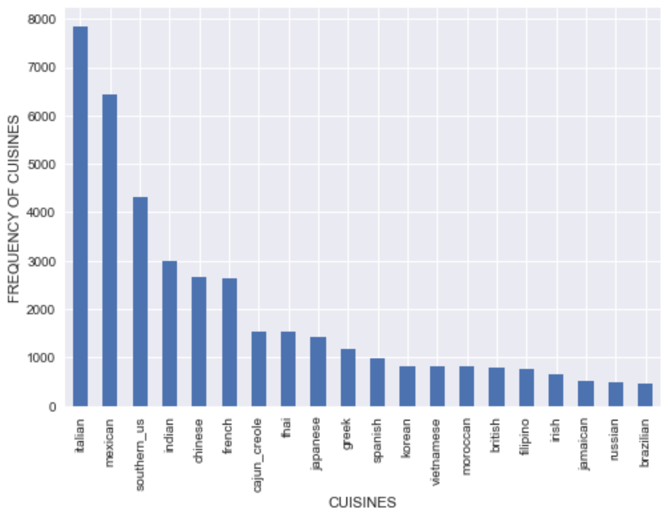

Fig. 3. Bar plot of Cuisines

them together to get a more accurate and satble prediction. Random Forest algorithm randomly selects observations and features to build several decision trees and then averages the results.Random Forest prevents overfitting most of the time, by creating random subsets of the features and building smaller trees using these subsets.

When we used Random Forests with CountVectorizer it gave an accuracy of 76.13%. But this model took a lot of time to fit and predict. This is the main limitation of Random Forest. A large number of trees can make the algorithm too slow and ineffective for real-time predictions. In general, these algorithms are fast to train, but quite slow to create predictions once they are trained. A more accurate prediction requires more trees, which results in a slower model.

We also tried to use SVM which gave an accuracy of 76.33%. This was followed by using Neural Networks as the next classification technique. We used MLPClassifier from the sklearn's available neural network. A multilayer perceptron (MLP) is a class of feedforward artificial neural network. An MLP consists of, at least, three layers of nodes: an input layer, a hidden layer and an output layer. MLP utilizes backpropogation for training. Its multiple layers and non-linear activation distinguishes MLP from a linear perceptron. MLP's are suitable for classification prediction problems where inputs are assigned a class or label. This reported an accuracy of 74.95 %.

3) DOC2VEC APPROACH: Doc2Vec is an extension to the Word2Vec-approach towards documents. Its intention is to encode (whole) documents consisting of lists of sentences, rather than lists of ungrouped sentences. doc2vec adds a placeholder input-neuron, fed by an unique value for each document, like the id or hash-value. According to the assumption behind, the placeholder should be able to represent some / more specifics of each particular document, and accomplishes this in a more appropriate / accurate way than the implicit document encoding in word2vec. The vectors generated by doc2vec can be used for tasks like finding similarity between sentences/paragraphs/documents or classification of documents.
We have implemented this using Doc2Vec paragraph embeddings is used from the gensim package to convert the Documents to Vectors. The ingredients in a recipe are concatenated with space and each of these strings are passed as a Document to the Doc2Vec module. To convert these documents to tagged documents, a function tag_docs() is built which takes a series of documents and its labels. This function returns the labelled documents. The function train_doc2vec_model() takes Labelled Documents returned by tag_docs() function and uses Doc2Vec function to train the doc2vec model.

To build the final feature vector for the classifier, the function vec_for_learning() takes as parameter, the training model and labelled documents returned by the train_doc2vec_model() and tag_docs() functions. The hashfxn infer_vector() is used to retrain the document vector. The reason for this being You could certainly use the vectors learned during training. But note that during much of their 20-training-passes, the model itself was still undergoing rapid change, and was far from its final state. Ive sometimes seen that re-inferred vectors, often work better for downstream tasks. This is perhaps because then all 20-inference-passes, across all re-inferred vectors, equally benefit from the same final frozen model state, quotedGordon Mohrfroma Google group QA. The vec_for_learning() function returns the tuple of target variable and feature variables which can be used to train different classification models.

Now, to use any classifier, pass the target and feature variable to tag_docs() function then input the return value of tag_docs to train_doc2vec_model() and finally the output of train_doc2vec_model() and tag_docs() is passed to the vec_for_learning() function and its return values are used to train the classification model and the trained model will be used for prediction. We started by implementing Logistic Regression. The main intution behind using the Logistic Regression is that the problem of predicting the cuisine from ingredients is a Multi-Class Text Classification problem and Logistic Regression can handle multiple classes. Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic/sigmoid function. A logistic regression model has the ability to generalize. The goal is to get a logistic regression model trained by the doc2vec feature that is able to classify unseen documents. Therefore, the models ability to generalize is important in this case We got an accuaracy of 73.06% from logistic regression. This was followed by SVM and Random Forests which reported an accuracy of 72.19% and 67.49% respectively.

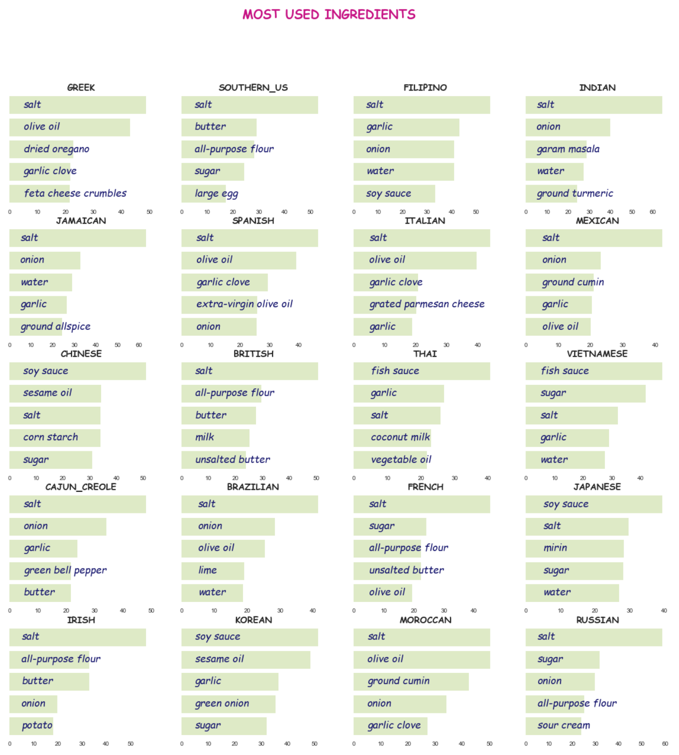

Fig. 4. Most used Ingredients

VI. CONCLUSION

Through the course of our project, our team made three different vectorized formats using TfidfVectorizer, CountVectorizer and Doc2Vec and put them into test with SVM, Logistic Regression, Random Forests, Multi-layer perceptron(MVP) neural network and Naive Bayes classification algorithms.

The results show that SVM yields the best result when tfidf vectorizer is used. One of the reason SVM gave the best accuracy with tfidf is because it takes into consideration not only the cuisine that occurs most frequently but also which is the principal ingredient with the help of idf and as mentioned SVM takes into consideration even the most varied type of data. (Inverse Document Frequency).

It was witnessed that the varied analysis of the ingredient set with different classification techniques gives an insight into the flavour pairings.The experimental results can be used in one of the most commercially viable aspect of improving and generating recipe prediction engines which can provide edible recipes from a given group of ingredients by automatically predicting the cuisine. Higher accuracies may be achieved by using various other Ensemble Methods (Voting Classifier and Bagged Trees) and generative models which we plan to implemet in future.

Vectorizer Model	**Classifier**	**Accuracy**
TfidfVectorizer	SVM	78.49 %
	Random Forests	75.01 %
	XGBoost	73.07 %
	MultinomialNB	71.98 %
CountVectorizer	SVM	76.09 %
	Random Forests	75.41 %
	Neural Net	74.19 %
Doc2Vec	Logistic Regression	73.06 %
	SVM	72.19 %
	Random Forests	67.49 %

Fig. 5. Accuracy of various models applied

VII. INDIVIDUAL CONTRIBUTIONS

A. *Siddharth Itagi*

1)For pre-processing of the data, WordLemmatization and WordStemming was done to remove accents from the words and remove words having the same meaning.
2)A Bar Chart was made specifying which cuisine has highest frquency.
3)Tf-idf Vectorizer was applied on textual data and then Random Forests, SVM and Naive Bayes classification models were used to predict the cuisine.

B. *Vaibhav Pundir*

1)Cleaning of data and stop-words, punctuation and digits were removed.
2)No. of distinct ingredients used in different cuisines using Bar Plot. Visualization of ”Most used ingredients in each cuisine” and ”Cuisine specific ingredient” using Horizontal Bar Charts.
3)doc2vec paragraph embedding to vectorize text and feed the transformed data to Random Forests, Logistic Regression and SVM to classify the ingredients.

C. *Krishna Khurana*

1)In the Data Visualization phase , a donought chart representing the usage of Top 10 ingredients of the whole dataset was shown.
2)Text Vectorization - CountVectorizer module of the scikitlearn was used to convert text data into a matrix form so can be fed to various Classification models.
3)After transforming raw text data(using CountVectorizer), SVM, Random Forest and multi layer perceptron neural network models were used to fit ingredients and predict the target cuisine for it.

REFERENCES

[1] Kevin K. Do , Duke University Durham, NC 27708, “Whats Cooking? Predicting Cuisines from Recipe Ingredients”
[2] Rishikesh Ghewari, Sunil Raiyani ”Predicting Cuisine from Ingredients”
[3] Shobha Jayaraman, Tanupriya Choudhury, Praveen Kumar-”Analysis of Classification models based on cuisine prediction using machine learning”-2017 International Conference On Smart Technology for Smart Nation

[4] R. M. Rahul Venkatesh Kumar , M. Anand Kumar and K. P. Soman, “Cuisine Prediction based on Ingredients using Tree